FIG.1

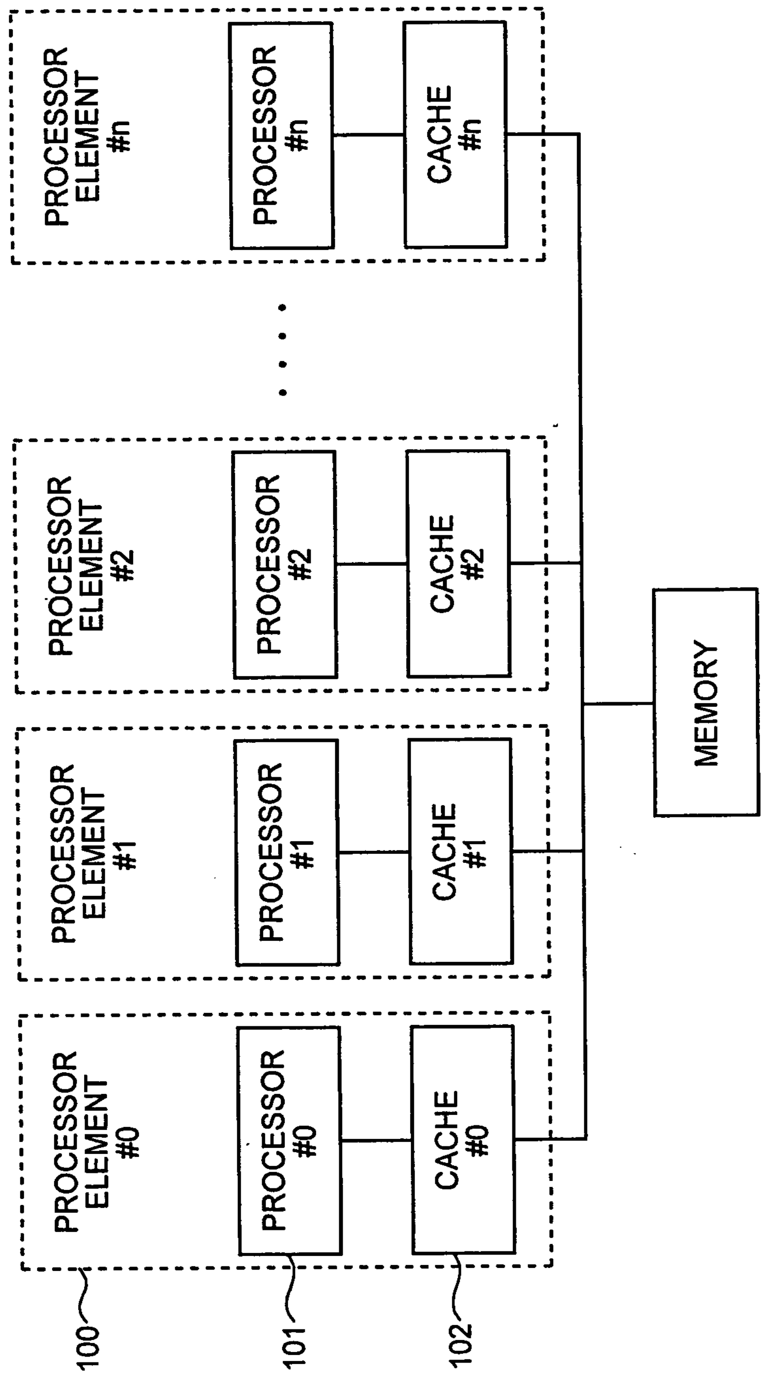

PROCESSOR ELEMENT #0
PROCESSOR #0
CACHE #0
PROCESSOR ELEMENT #1
PROCESSOR #1
CACHE #1
PROCESSOR ELEMENT #2
PROCESSOR #2
CACHE #2
PROCESSOR ELEMENT #n
PROCESSOR #n
CACHE #n
MEMORY
100
101
102

FIG.2

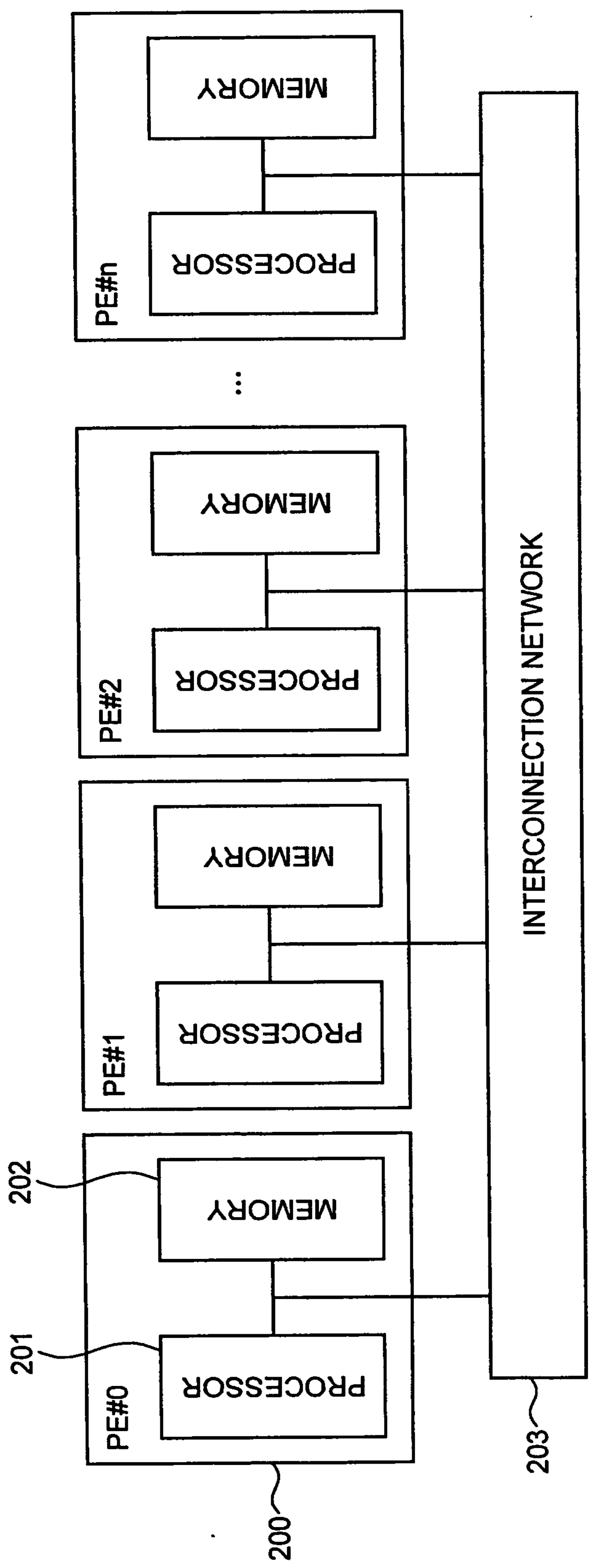

200
PE#0
PROCESSOR
201
MEMORY
202
PE#1
PROCESSOR
MEMORY
PE#2
PROCESSOR
MEMORY
...
PE#n
PROCESSOR
MEMORY
INTERCONNECTION NETWORK
203

FIG.3

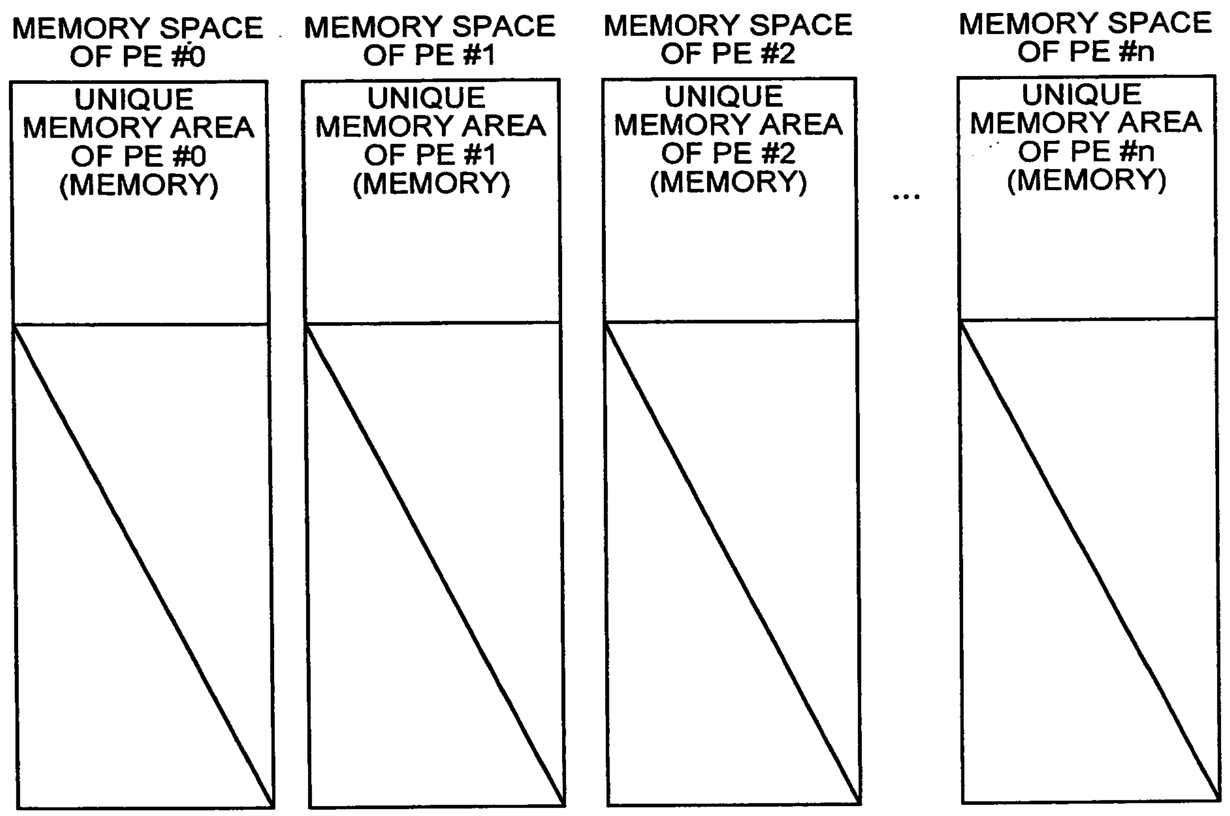

MEMORY SPACE OF PE #0
MEMORY SPACE OF PE #1
MEMORY SPACE OF PE #2
MEMORY SPACE OF PE #n
UNIQUE MEMORY AREA OF PE #0 (MEMORY)
UNIQUE MEMORY AREA OF PE #1 (MEMORY)
UNIQUE MEMORY AREA OF PE #2 (MEMORY)
...
UNIQUE MEMORY AREA OF PE #n (MEMORY)

FIG.4

```
#include <stdio.h>
#include <string.h>
#include "mpi.h"

int
main(int argc, char ＊＊argv)
{
      int my_rank;／＊RANK OF CURRENT PROCESS＊／
      int source;／＊RANK OF TRANSMISSION PROCESS＊／
      int dest;／＊RANK OF RECEIVING PROCESS＊／
      int tag=0;／＊MESSAGE TAG＊／
      char message[100];／＊STORAGE PLACE OF MESSAGE＊／
      MPI_Status status;／＊RETURN STATUS OF RECEPTION＊／

      ／＊MPI START UP＊／
      MPI_Init (&argc, argv) ;

      ／＊REQUEST RANK OF CURRENT PROCESS＊／
      MPI_Comm_rank (MPI_COMM_WORLD. &my_rank) ;

      if (my_rank ! =0) {
        ／＊MESSAGE CREATION＊／
        sprintf (message, "Greentings from process %d＼n", my_rank) ;
        dest=0 ;
        ／＊' ＼USE strlen+1 as 0' is also sent＊／
        MPI_Send (message. strlen (message)+1, MPI_CHAR, dest, tag,
        MPI_COMM_WORLD) ;
      } else {
        source=1;
        MPI_Recv (message, sizeof(message), MPI_CHAR, source,tag,
        MPI_COMM_WORLD, &status) ;
        printf ("%s ＼n" , message) ;
        }

        ／＊MPI SHUT DOWN＊／
        MPI_Finalize () ;
        return 0 ;
}
```

FIG.5

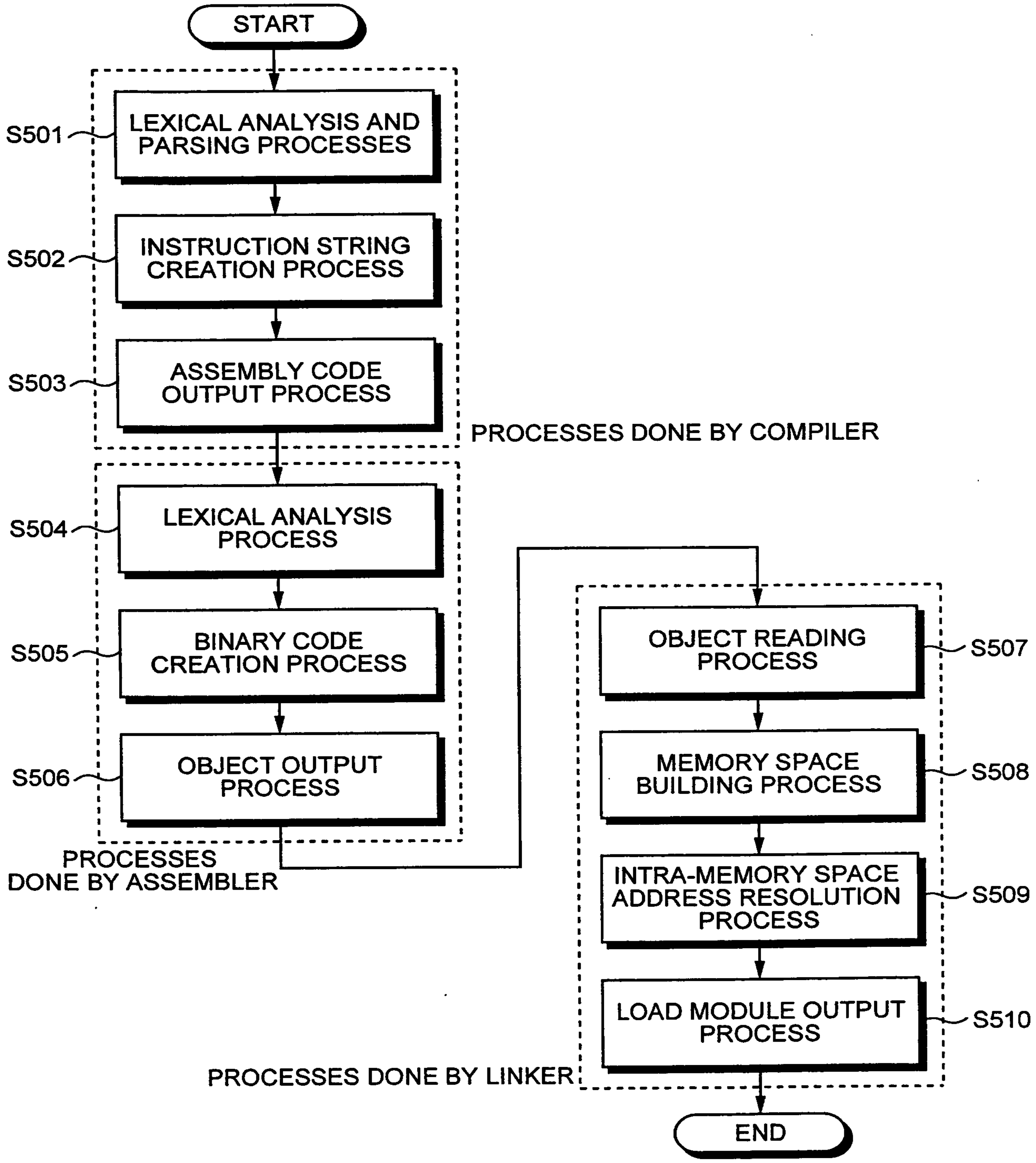

START
S501
LEXICAL ANALYSIS AND PARSING PROCESSES
S502
INSTRUCTION STRING CREATION PROCESS
S503
ASSEMBLY CODE OUTPUT PROCESS
PROCESSES DONE BY COMPILER
S504
LEXICAL ANALYSIS PROCESS
S505
BINARY CODE CREATION PROCESS
S506
OBJECT OUTPUT PROCESS
PROCESSES DONE BY ASSEMBLER
OBJECT READING PROCESS
S507
MEMORY SPACE BUILDING PROCESS
S508
INTRA-MEMORY SPACE ADDRESS RESOLUTION PROCESS
S509
LOAD MODULE OUTPUT PROCESS
S510
PROCESSES DONE BY LINKER
END

FIG.6

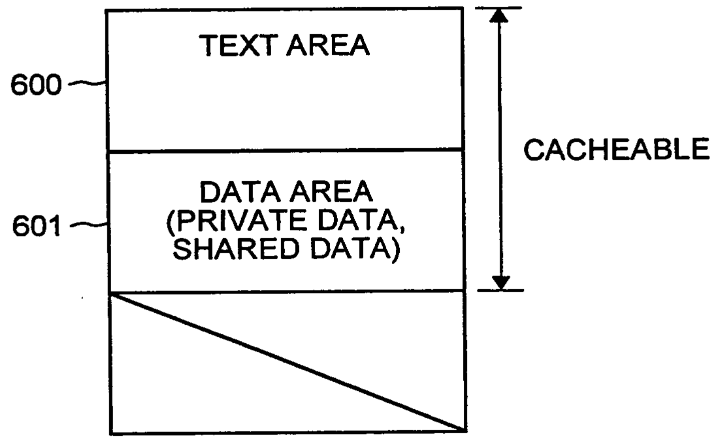

TEXT AREA
600
DATA AREA (PRIVATE DATA, SHARED DATA)
601
CACHEABLE

FIG.7

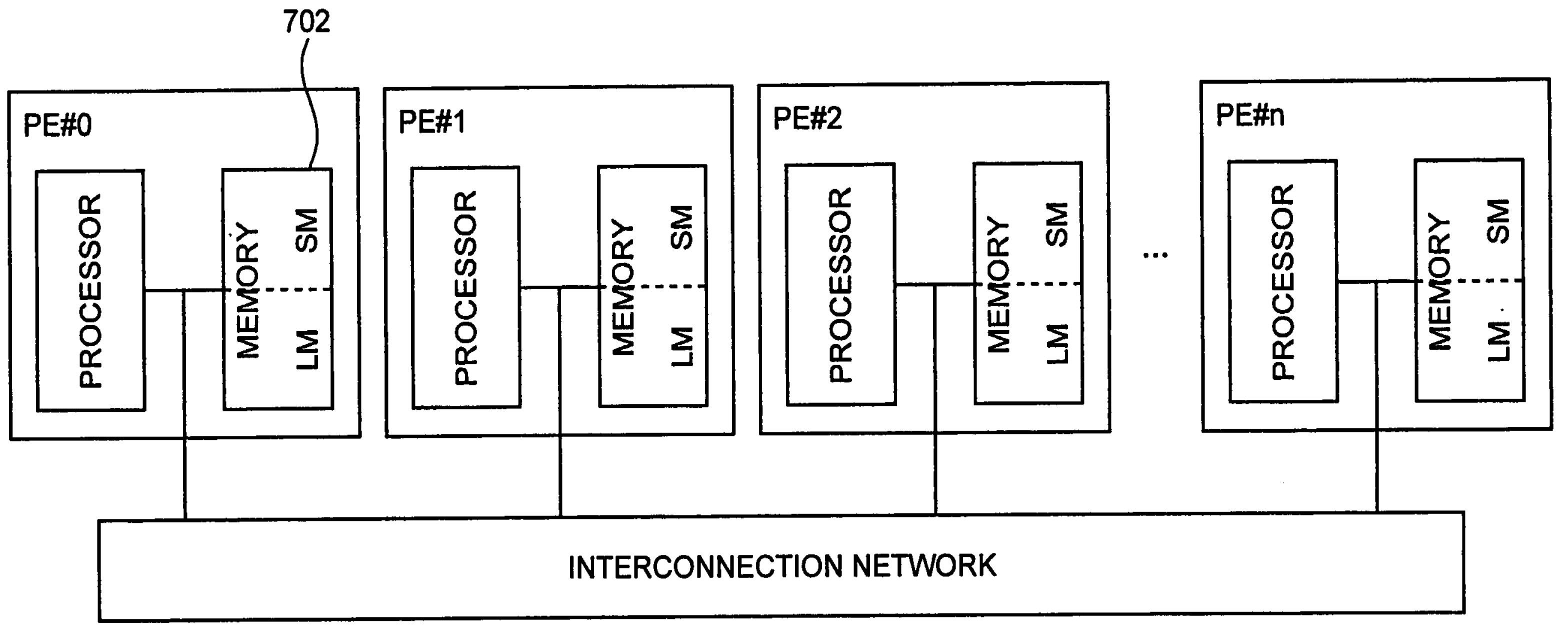

702
PE#0
PROCESSOR
MEMORY
SM
LM
PE#1
PROCESSOR
MEMORY
SM
LM
PE#2
PROCESSOR
MEMORY
SM
LM
...
PE#n
PROCESSOR
MEMORY
SM
LM
INTERCONNECTION NETWORK

FIG.8

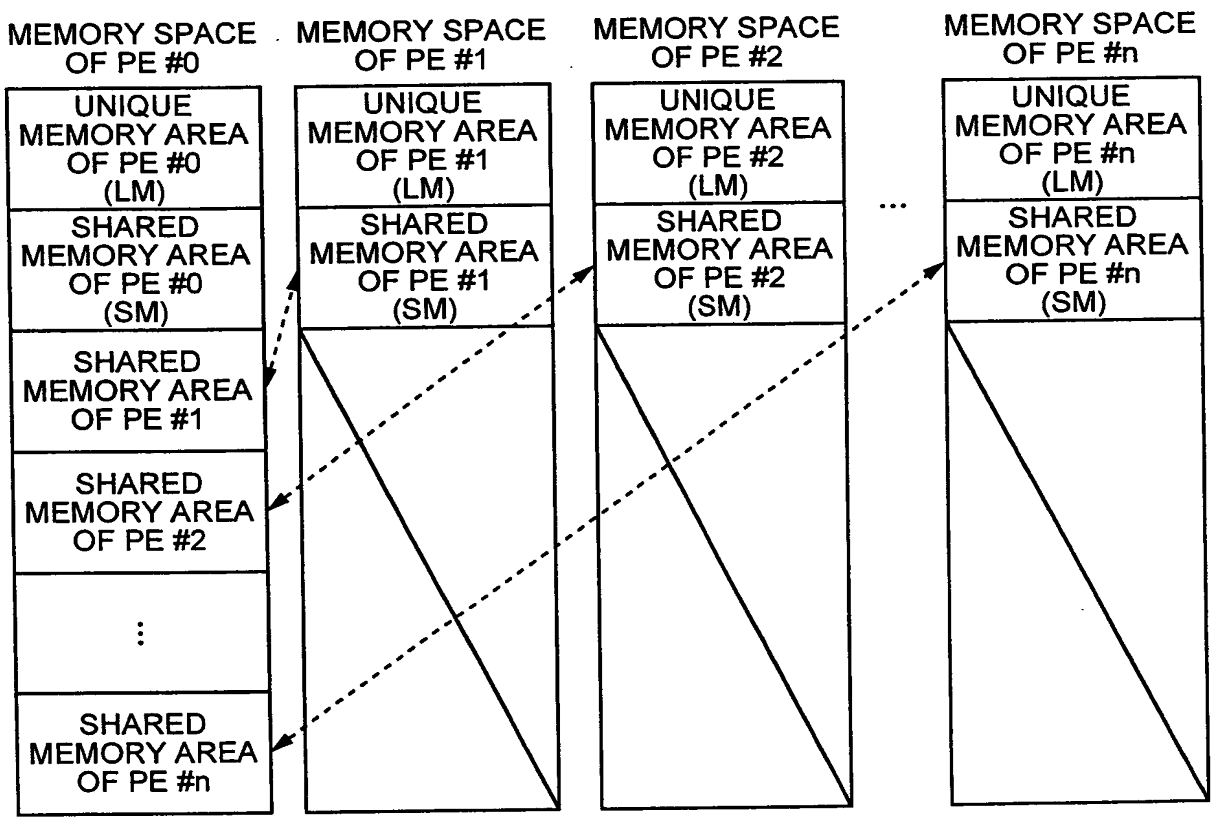

MEMORY SPACE OF PE #0
MEMORY SPACE OF PE #1
MEMORY SPACE OF PE #2
MEMORY SPACE OF PE #n
UNIQUE MEMORY AREA OF PE #0 (LM)
SHARED MEMORY AREA OF PE #0 (SM)
SHARED MEMORY AREA OF PE #1
SHARED MEMORY AREA OF PE #2
⋮
SHARED MEMORY AREA OF PE #n
UNIQUE MEMORY AREA OF PE #1 (LM)
SHARED MEMORY AREA OF PE #1 (SM)
UNIQUE MEMORY AREA OF PE #2 (LM)
SHARED MEMORY AREA OF PE #2 (SM)
...
UNIQUE MEMORY AREA OF PE #n (LM)
SHARED MEMORY AREA OF PE #n (SM)

FIG.9

```
#include <stdio.h>
#include <string.h>
#include "mpi.h"

int
main(int argc, char **argv)
{
      int my_rank;/*RANK OF CURRENT PROCESS*/
      int source;/*RANK OF TRANSMISSION PROCESS*/
      int tag=0;/*MESSAGE TAG*/
      char message[100];/*STORAGE PLACE OF MESSAGE*/
      MPI_Status status;/*RETURN STATUS OF RECEPTION*/

      /*MPI START UP*/
      MPI_Init (&argc, argv) ;

      /*REQUEST RANK OF CURRENT PROCESS*/
      MPI_Comm_rank (MPI_COMM_WORLD. &my_rank) ;

      source=1;
      MPI_Recv (message, sizeof(message), MPI_CHAR, source,tag,
      MPI_COMM_WORLD, &status) ;
      printf ("%s\n", message) ;

      /*MPI SHUT DOWN*/
      MPI_Finalize () ;
      return 0 ;
}
```

FIG.10

```
#include <stdio.h>
#include <string.h>
#include “mpi.h”

int
main(int argc, char ＊＊argv)
{
      int my_rank;／＊RANK OF CURRENT PROCESS＊／
      int dest;／＊RANK OF RECEIVING PROCESS＊／
      int tag=0;／＊MESSAGE TAG＊／
      char message[100];／＊STORAGE PLACE OF MESSAGE＊／

      ／＊MPI START UP＊／
      MPI_Init (&argc, argv) ;

      ／＊REQUEST RANK OF CURRENT PROCESS＊／
      MPI_Comm_rank (MPI_COMM_WORLD. &my_rank) ;

      ／＊MESSAGE CREATION＊／
      sprintf (message, “Greentings from process %d＼n”, my_rank) ;
      dest=0 ;
      ／＊’＼USE strlen +1 as 0’is also sent＊／
      MPI_Send (message. strlen (message)+1, MPI_CHAR, dest, tag,
      MPI_COMM_WORLD) ;

      ／＊MPI SHUT DOWN＊／
      MPI_Finalize () ;
      return 0 ;
}
```

FIG.11

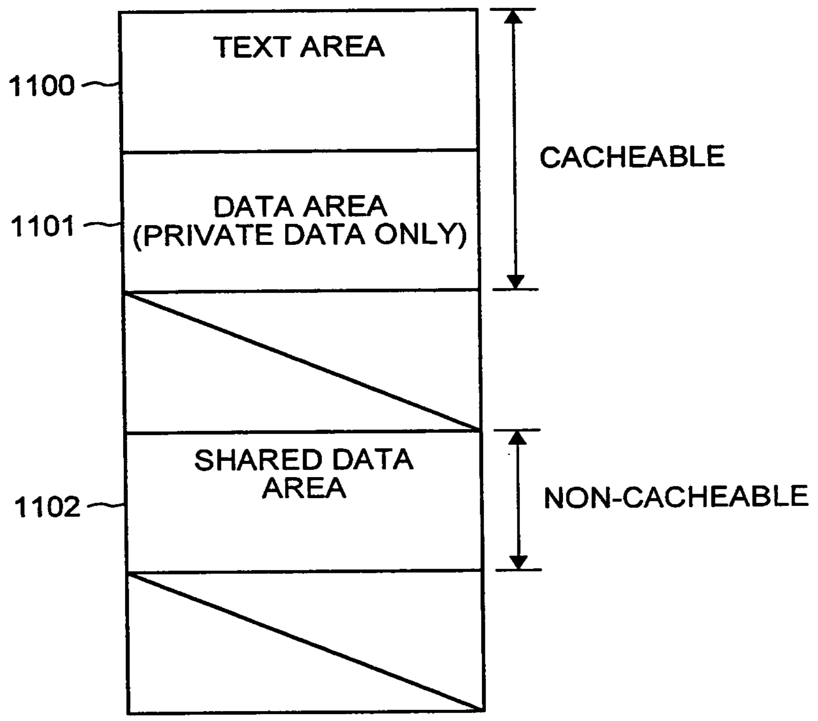

TEXT AREA
1100
DATA AREA (PRIVATE DATA ONLY)
1101
CACHEABLE
SHARED DATA AREA
1102
NON-CACHEABLE

FIG.12

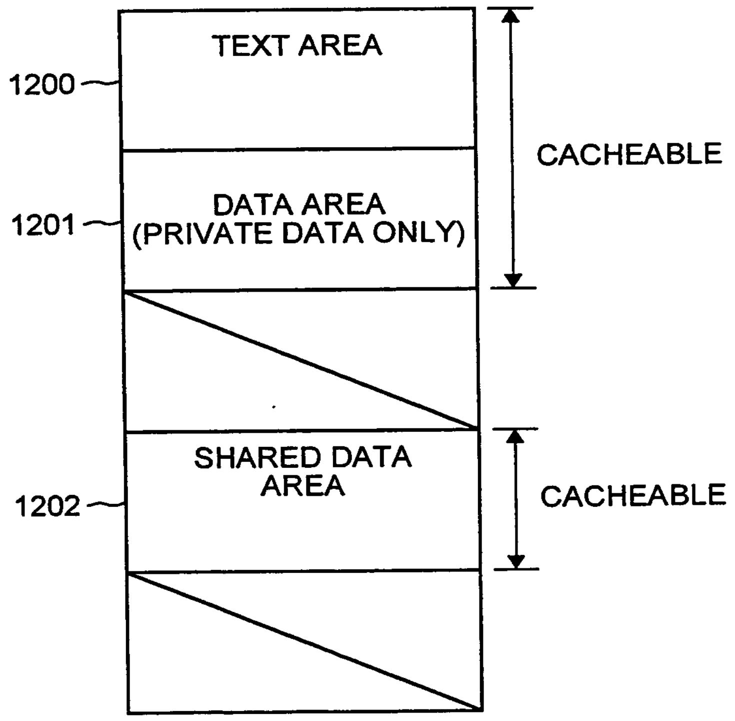

TEXT AREA
1200
DATA AREA (PRIVATE DATA ONLY)
1201
CACHEABLE
SHARED DATA AREA
1202
CACHEABLE

FIG.13

```
int input;
int output;
extern int in;
extern int out;


void
Th0(void)
{
        MOVE(&in, &input, sizeof (in));          /*Th0-1*/
        START(1."Th1" );                         /*Th0-2*/
        MOVE(&output, &out, sizeof (output));/*Th0-3*/
}
```

FIG.14

```
int in;
int out;


void
Th1(void)
{
        extern void f1(int*, int*);

        f1(&in, &out);                           /*Th1-1*/
}
```

FIG.15

	MEMORY SPACE OF PE #0		MEMORY SPACE OF PE #1	
	ADDRESS	CONTENTS	ADDRESS	CONTENTS

TEXT AREA	0x0000	void
		Th0(void)
		{
		MOVE(0x3000, 0x1000, sizeof(in)) ;
		START(1."Th1") ;
		MOVE(0x1004, 0x3004, sizeof(output)) ;
		}
			0x0000	void
				Th1(void)
				{
				f1(0x2000, 0x2004));
				}
				
				void
				f1(int ∗in, int, ∗out)
				{
				:
				:
				}

DATA AREA	0x1000	int input;	0x1000	
	0x1004	int output;		
SHARED DATA AREA #0	0x2000			
SHARED DATA AREA #1	0x3000	int in;	0x2000	int in;
	0x3004	int out;	0x2004	int out;

FIG.16

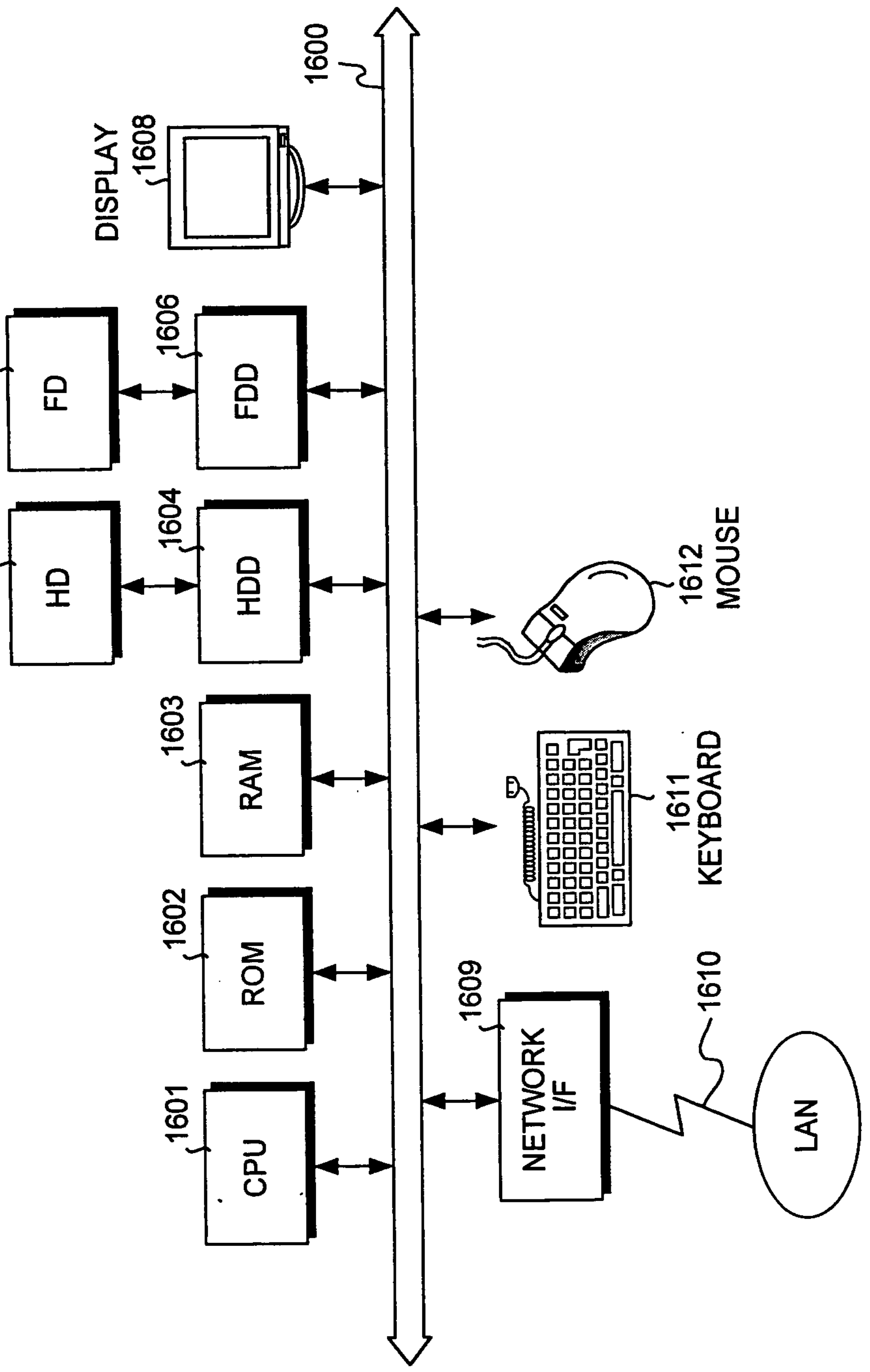

1600
CPU
1601
ROM
1602
RAM
1603
HDD
1604
HD
1605
FDD
1606
FD
1607
DISPLAY
1608
NETWORK I/F
1609
1610
LAN
KEYBOARD
1611
MOUSE
1612

FIG.17

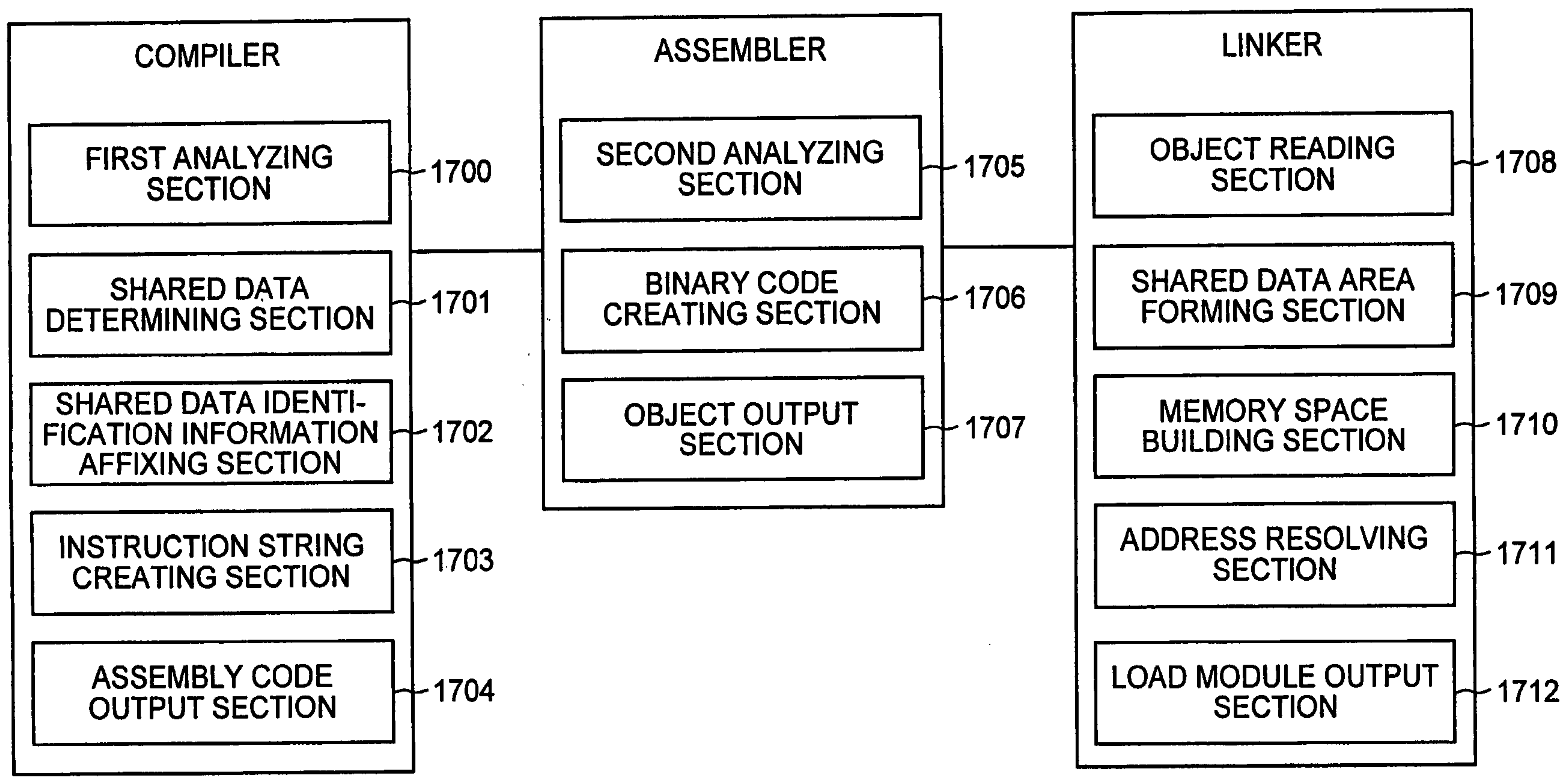

COMPILER
FIRST ANALYZING SECTION
1700
SHARED DATA DETERMINING SECTION
1701
SHARED DATA IDENTI-FICATION INFORMATION AFFIXING SECTION
1702
INSTRUCTION STRING CREATING SECTION
1703
ASSEMBLY CODE OUTPUT SECTION
1704
ASSEMBLER
SECOND ANALYZING SECTION
1705
BINARY CODE CREATING SECTION
1706
OBJECT OUTPUT SECTION
1707
LINKER
OBJECT READING SECTION
1708
SHARED DATA AREA FORMING SECTION
1709
MEMORY SPACE BUILDING SECTION
1710
ADDRESS RESOLVING SECTION
1711
LOAD MODULE OUTPUT SECTION
1712

FIG.18

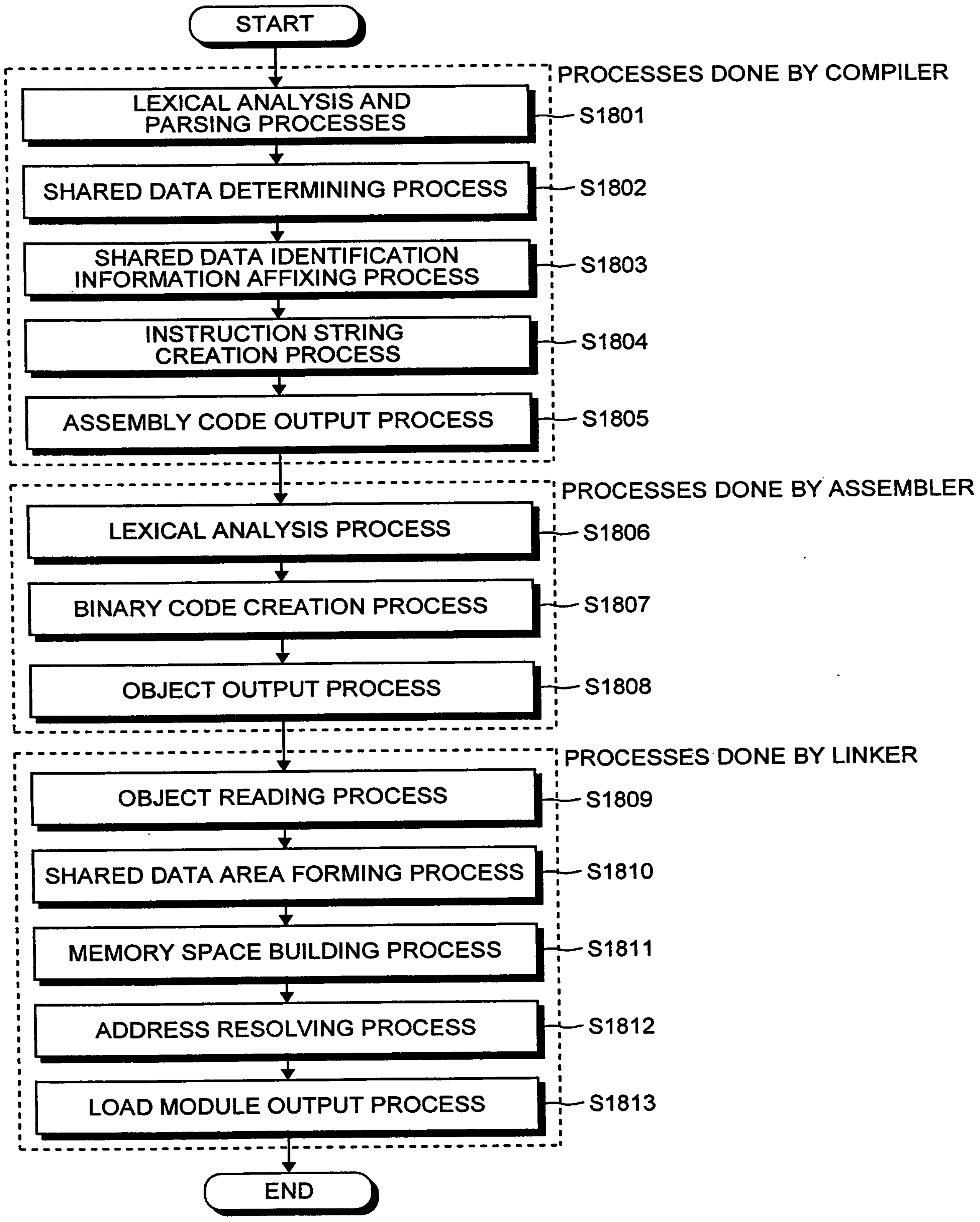

START
PROCESSES DONE BY COMPILER
LEXICAL ANALYSIS AND PARSING PROCESSES
S1801
SHARED DATA DETERMINING PROCESS
S1802
SHARED DATA IDENTIFICATION INFORMATION AFFIXING PROCESS
S1803
INSTRUCTION STRING CREATION PROCESS
S1804
ASSEMBLY CODE OUTPUT PROCESS
S1805
PROCESSES DONE BY ASSEMBLER
LEXICAL ANALYSIS PROCESS
S1806
BINARY CODE CREATION PROCESS
S1807
OBJECT OUTPUT PROCESS
S1808
PROCESSES DONE BY LINKER
OBJECT READING PROCESS
S1809
SHARED DATA AREA FORMING PROCESS
S1810
MEMORY SPACE BUILDING PROCESS
S1811
ADDRESS RESOLVING PROCESS
S1812
LOAD MODULE OUTPUT PROCESS
S1813
END

FIG.19

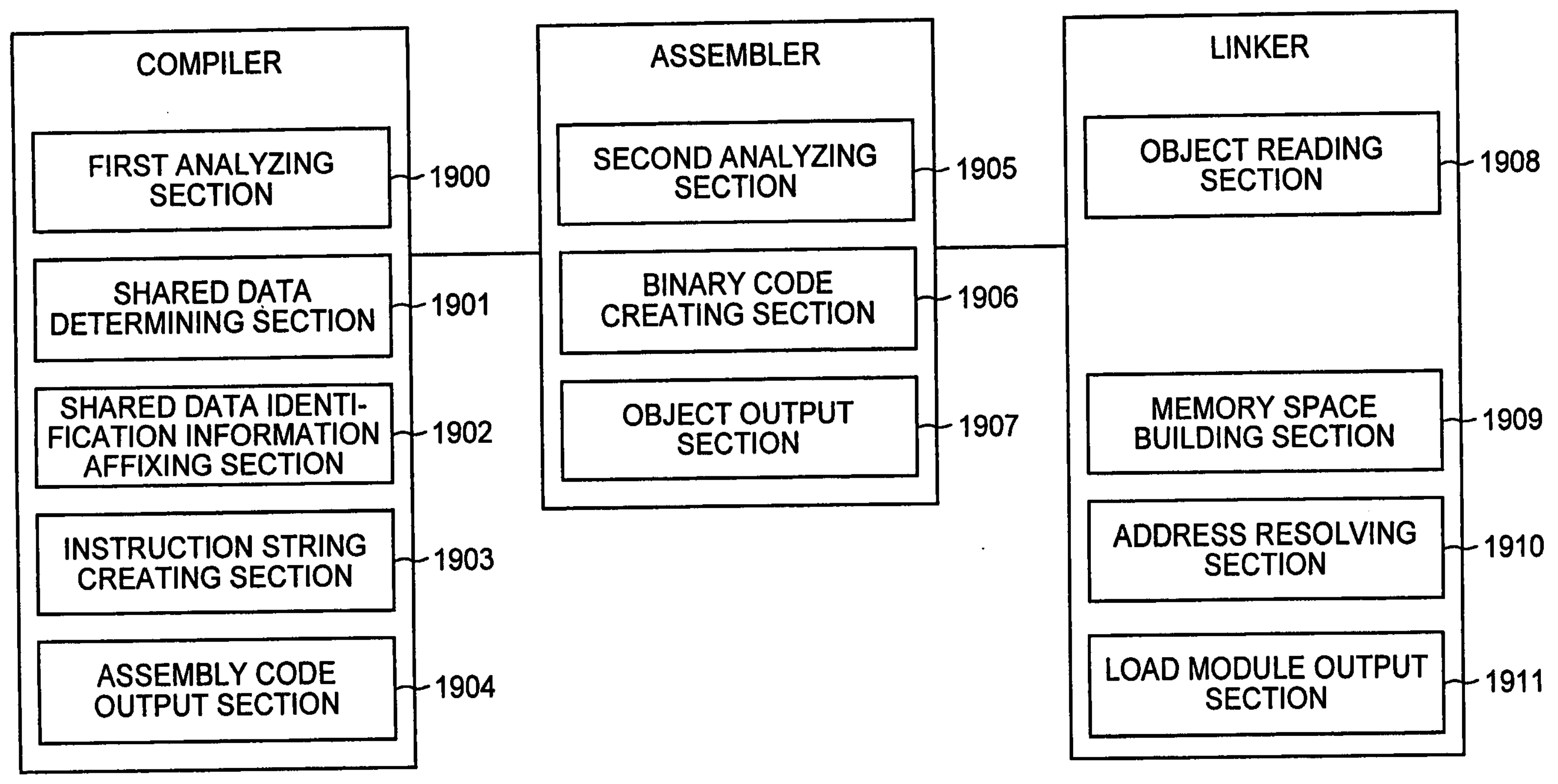

COMPILER
FIRST ANALYZING SECTION
1900
SHARED DATA DETERMINING SECTION
1901
SHARED DATA IDENTI-FICATION INFORMATION AFFIXING SECTION
1902
INSTRUCTION STRING CREATING SECTION
1903
ASSEMBLY CODE OUTPUT SECTION
1904
ASSEMBLER
SECOND ANALYZING SECTION
1905
BINARY CODE CREATING SECTION
1906
OBJECT OUTPUT SECTION
1907
LINKER
OBJECT READING SECTION
1908
MEMORY SPACE BUILDING SECTION
1909
ADDRESS RESOLVING SECTION
1910
LOAD MODULE OUTPUT SECTION
1911

FIG.20

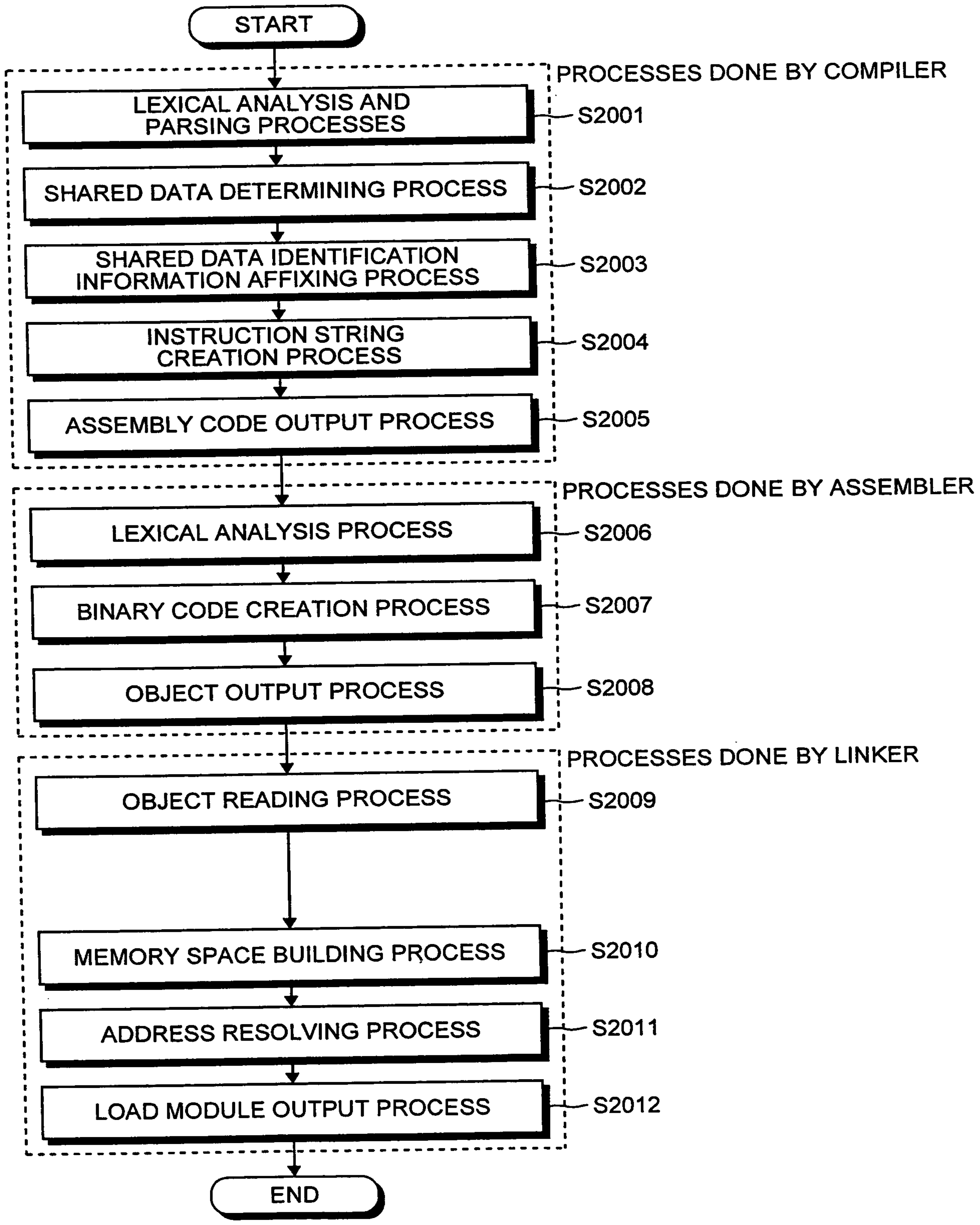

START
PROCESSES DONE BY COMPILER
LEXICAL ANALYSIS AND PARSING PROCESSES
S2001
SHARED DATA DETERMINING PROCESS
S2002
SHARED DATA IDENTIFICATION INFORMATION AFFIXING PROCESS
S2003
INSTRUCTION STRING CREATION PROCESS
S2004
ASSEMBLY CODE OUTPUT PROCESS
S2005
PROCESSES DONE BY ASSEMBLER
LEXICAL ANALYSIS PROCESS
S2006
BINARY CODE CREATION PROCESS
S2007
OBJECT OUTPUT PROCESS
S2008
PROCESSES DONE BY LINKER
OBJECT READING PROCESS
S2009
MEMORY SPACE BUILDING PROCESS
S2010
ADDRESS RESOLVING PROCESS
S2011
LOAD MODULE OUTPUT PROCESS
S2012
END

FIG.21

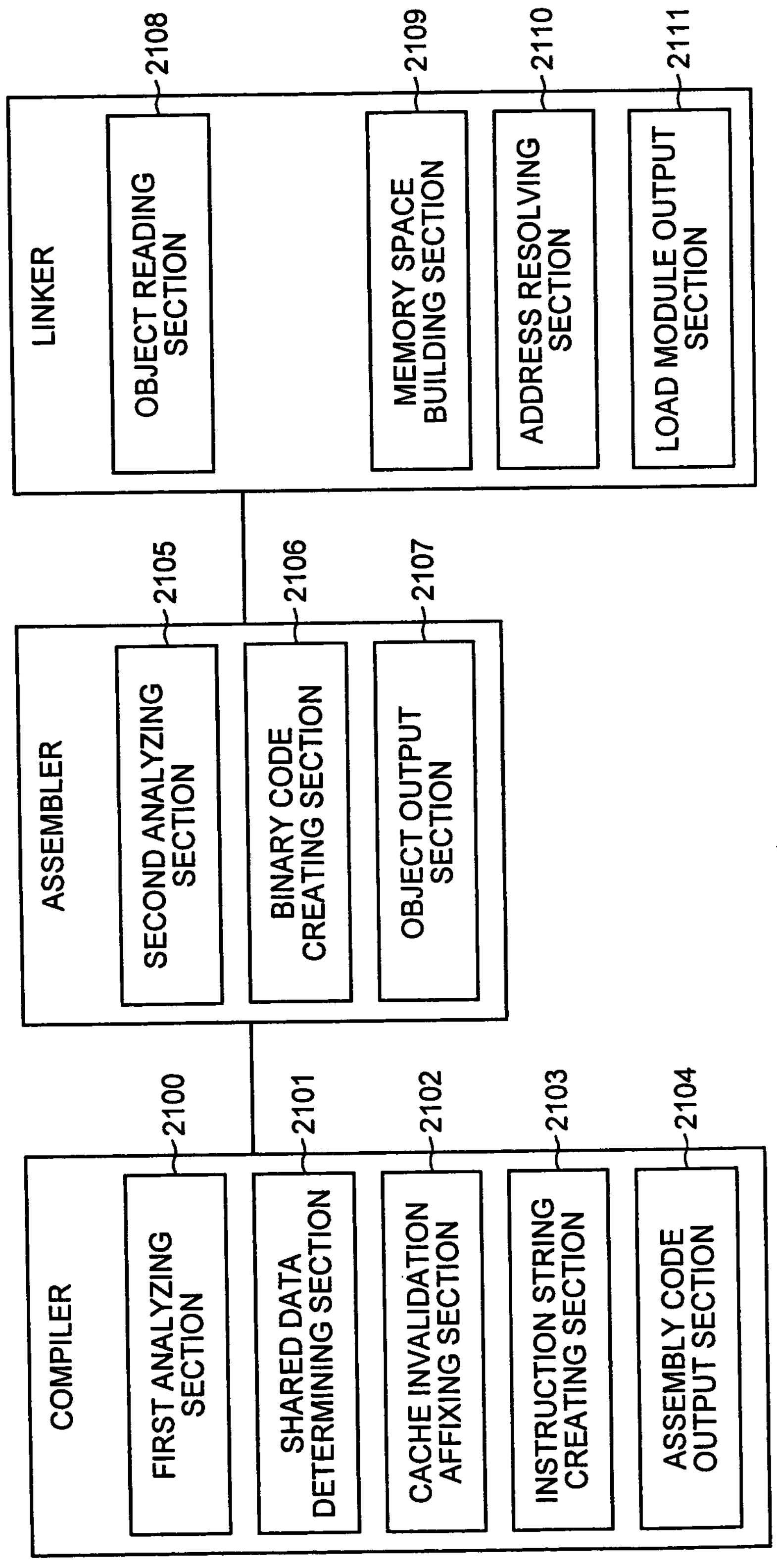

COMPILER
FIRST ANALYZING SECTION
2100
SHARED DATA DETERMINING SECTION
2101
CACHE INVALIDATION AFFIXING SECTION
2102
INSTRUCTION STRING CREATING SECTION
2103
ASSEMBLY CODE OUTPUT SECTION
2104
ASSEMBLER
SECOND ANALYZING SECTION
2105
BINARY CODE CREATING SECTION
2106
OBJECT OUTPUT SECTION
2107
LINKER
OBJECT READING SECTION
2108
MEMORY SPACE BUILDING SECTION
2109
ADDRESS RESOLVING SECTION
2110
LOAD MODULE OUTPUT SECTION
2111

FIG.22

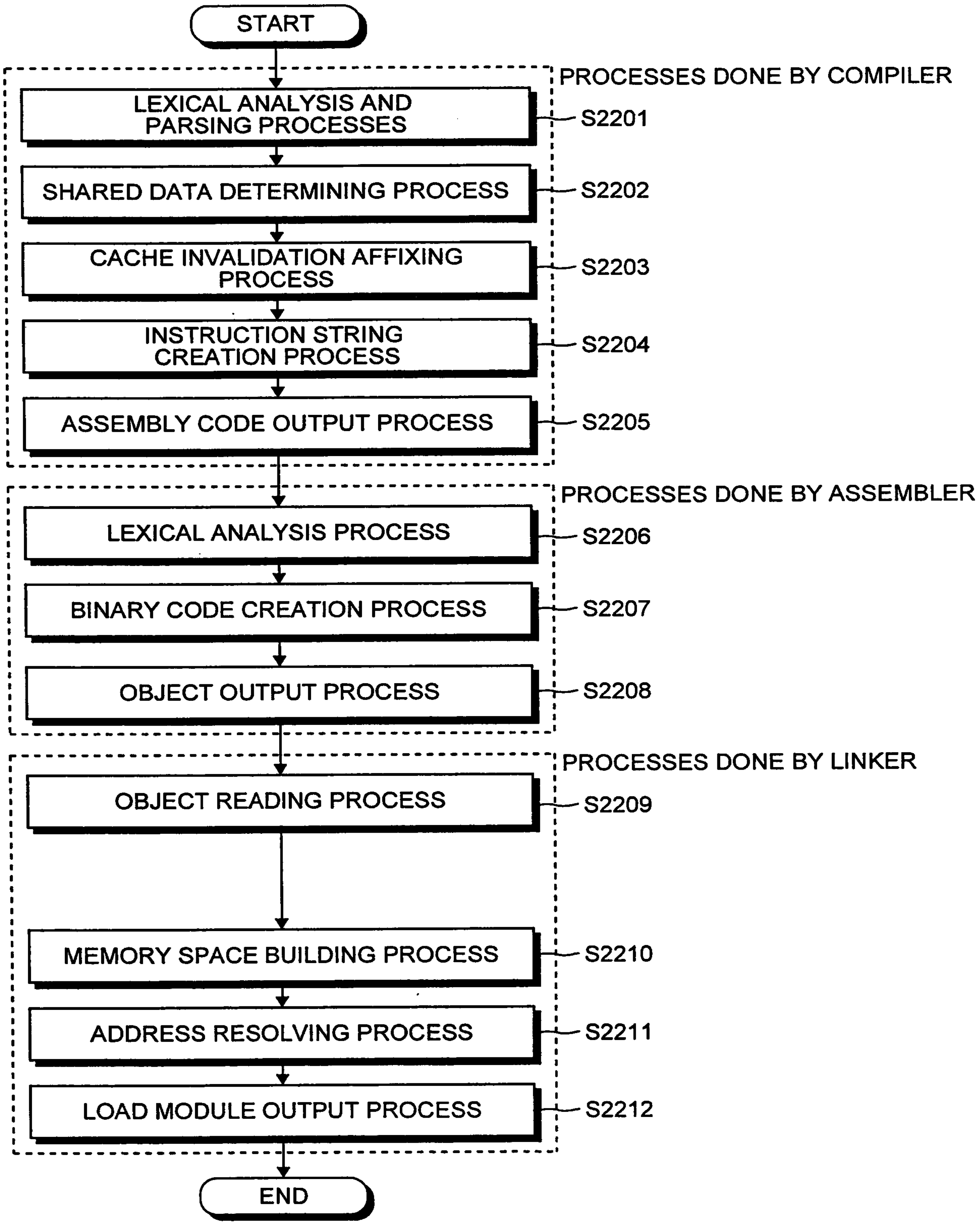

START
PROCESSES DONE BY COMPILER
LEXICAL ANALYSIS AND PARSING PROCESSES
S2201
SHARED DATA DETERMINING PROCESS
S2202
CACHE INVALIDATION AFFIXING PROCESS
S2203
INSTRUCTION STRING CREATION PROCESS
S2204
ASSEMBLY CODE OUTPUT PROCESS
S2205
PROCESSES DONE BY ASSEMBLER
LEXICAL ANALYSIS PROCESS
S2206
BINARY CODE CREATION PROCESS
S2207
OBJECT OUTPUT PROCESS
S2208
PROCESSES DONE BY LINKER
OBJECT READING PROCESS
S2209
MEMORY SPACE BUILDING PROCESS
S2210
ADDRESS RESOLVING PROCESS
S2211
LOAD MODULE OUTPUT PROCESS
S2212
END

FIG.23

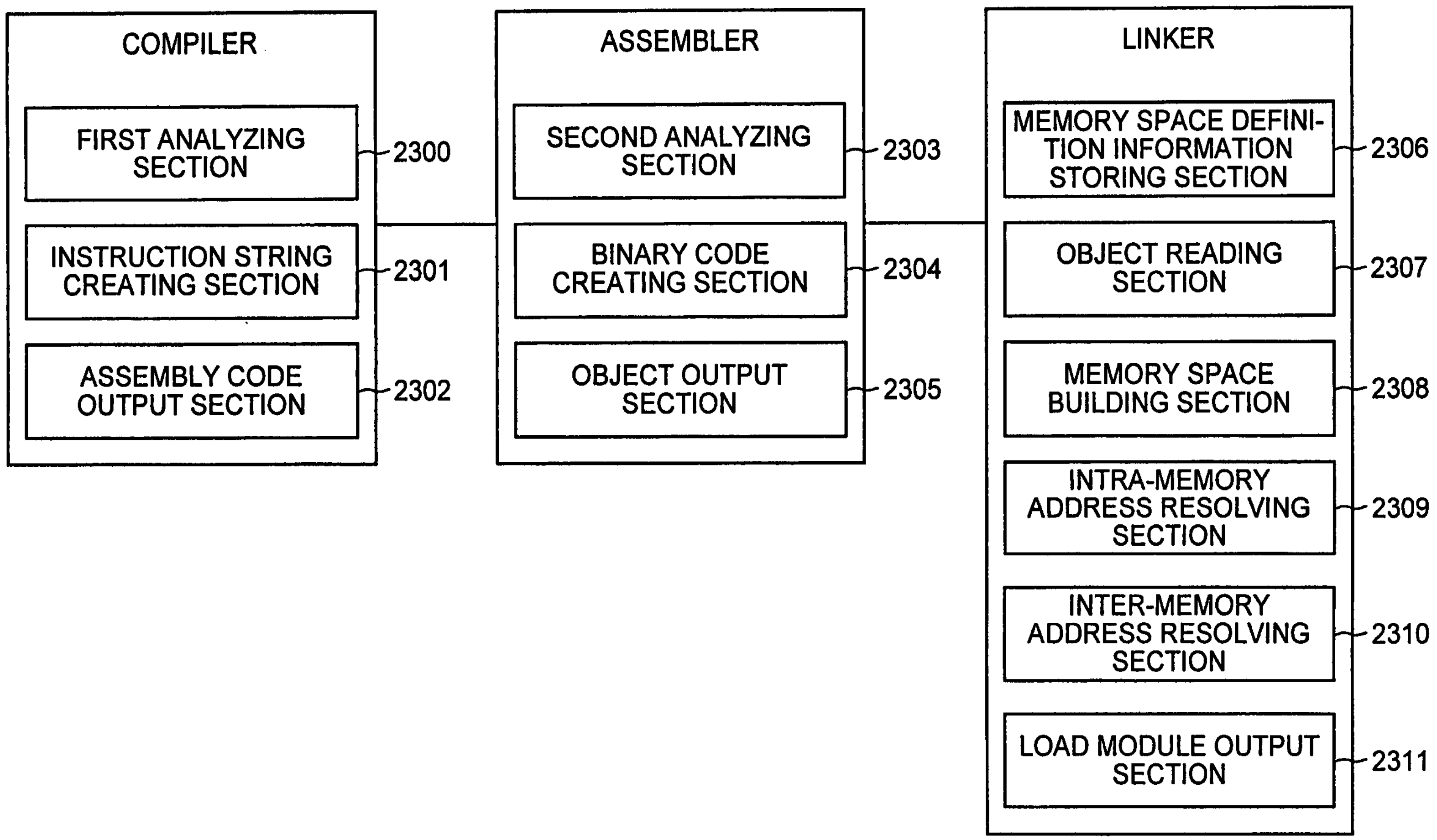

COMPILER
FIRST ANALYZING SECTION
2300
INSTRUCTION STRING CREATING SECTION
2301
ASSEMBLY CODE OUTPUT SECTION
2302
ASSEMBLER
SECOND ANALYZING SECTION
2303
BINARY CODE CREATING SECTION
2304
OBJECT OUTPUT SECTION
2305
LINKER
MEMORY SPACE DEFINITION INFORMATION STORING SECTION
2306
OBJECT READING SECTION
2307
MEMORY SPACE BUILDING SECTION
2308
INTRA-MEMORY ADDRESS RESOLVING SECTION
2309
INTER-MEMORY ADDRESS RESOLVING SECTION
2310
LOAD MODULE OUTPUT SECTION
2311

FIG.24

PE IDENTIFICATION	AREA NAME	STARTING ADDRESS	ENDING ADDRESS
PE #0	TEXT AREA	0x0000	0x0fff
	DATA AREA	0x1000	0x1fff
	SHARED DATA AREA #0	0x2000	0x2fff
	SHARED DATA AREA #1	0x3000	0x3fff
PE #1	TEXT AREA	0x0000	0x0fff
	DATA AREA	0x1000	0x1fff
	SHARED DATA AREA #1	0x2000	0x2fff

FIG.25

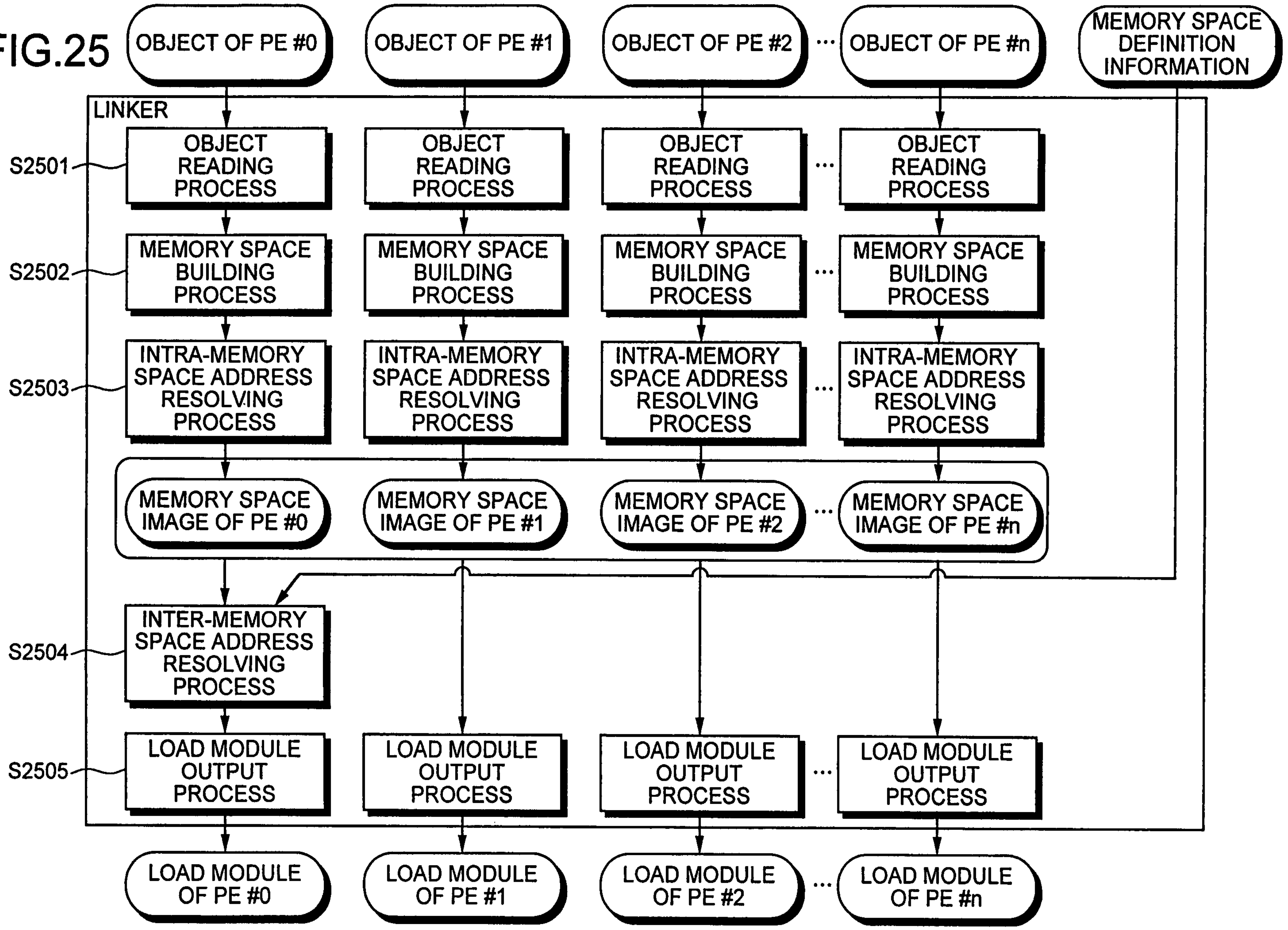
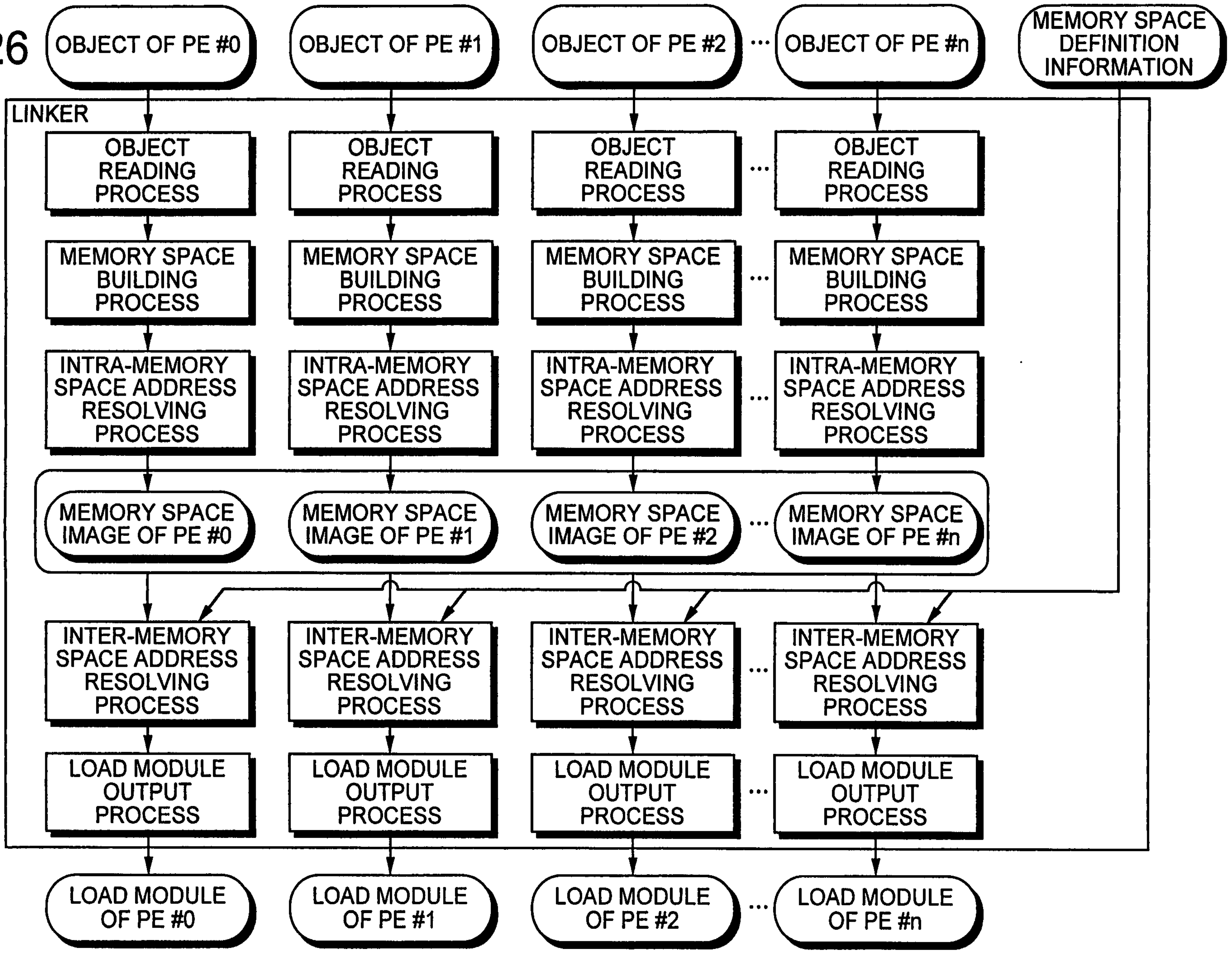

OBJECT OF PE #0
OBJECT OF PE #1
OBJECT OF PE #2
OBJECT OF PE #n
MEMORY SPACE DEFINITION INFORMATION
LINKER
S2501
OBJECT READING PROCESS
S2502
MEMORY SPACE BUILDING PROCESS
S2503
INTRA-MEMORY SPACE ADDRESS RESOLVING PROCESS
MEMORY SPACE IMAGE OF PE #0
MEMORY SPACE IMAGE OF PE #1
MEMORY SPACE IMAGE OF PE #2
MEMORY SPACE IMAGE OF PE #n
S2504
INTER-MEMORY SPACE ADDRESS RESOLVING PROCESS
S2505
LOAD MODULE OUTPUT PROCESS
LOAD MODULE OF PE #0
LOAD MODULE OF PE #1
LOAD MODULE OF PE #2
LOAD MODULE OF PE #n

FIG.26


OBJECT OF PE #0
OBJECT OF PE #1
OBJECT OF PE #2
OBJECT OF PE #n
MEMORY SPACE DEFINITION INFORMATION
LINKER
OBJECT READING PROCESS
MEMORY SPACE BUILDING PROCESS
INTRA-MEMORY SPACE ADDRESS RESOLVING PROCESS
MEMORY SPACE IMAGE OF PE #0
MEMORY SPACE IMAGE OF PE #1
MEMORY SPACE IMAGE OF PE #2
MEMORY SPACE IMAGE OF PE #n
INTER-MEMORY SPACE ADDRESS RESOLVING PROCESS
LOAD MODULE OUTPUT PROCESS
LOAD MODULE OF PE #0
LOAD MODULE OF PE #1
LOAD MODULE OF PE #2
LOAD MODULE OF PE #n